(12) **EUROPEAN PATENT SPECIFICATION**

(54) **Method for optimizing the motion of a multi-axis robot**

Verfahren zur Optimierung der Bewegung eines Mehr-Achsen-Roboters

Méthode d'optimalisation du mouvement d'un robot avec des axes multiples

(72) Inventor: Batchelder, John Sam
Somers, New York 10589 (US)

EP 0 642 893 B1

Description

FIELD OF THE INVENTION:

5   [0001]   This invention relates generally to the control of motors and, in particular, to the control of stepper motors such as those employed in robotics and rapid prototyping applications.

BACKGROUND OF THE INVENTION:

10   [0002]   High performance robotic systems may use closed loop control such as position, velocity, and acceleration sensors, feedback and feed forward calculations, and signal filtration. The use of feedback enables a level of performance near optimal levels of speed and accuracy for point-to-point moves.
   [0003]   As an example, a robotic system that is adapted for a rapid prototyping system (RPS), and that employs feedback to control a position of an extrusion nozzle in an x-y-z coordinate reference frame is disclosed in US-A-5,303,141
15   published on 12.04.1994 which does however not form part of the prior art.
   [0004]   An example of the generally applied use of feedback in robotics, through the use of encoders and possibly tachometers, is shown in US-A-4,749,347.
   [0005]   However, for low cost robot actuators the additional cost of providing feedback sensors and servo amplifiers may be prohibitive.
20   [0006]   Synchronization of multiple axis motion may also be required in robotic applications. Master/slave techniques are typically used, such as described in "Multi-axis Controller", by R.C. Russell, US-A-4,415,967.
   [0007]   The closest prior art EP-A-0440805 discloses generally a method according to the preamble of claim 1, for controlling the feed speed of a tool moving along a machining path in accordance with a machining programm. The method derives a correct feed speed.
25   [0008]   Optimization of a point-to-point move time may not, however, optimize the total move time over a trajectory. In an application of particular interest, the problem that is addressed by this invention can be understood by comparing two simple trajectories.
   [0009]   In a first case, 1000 line segments each 2.54×10^{-4}m (0.01 inches) long are concatenated to form a circle with a diameter of 8.08×10^{-2}m (3.18 inches). Ignoring truncation effects due to the finite step size possible along each axis,
30   the maximum speed that a robot should make this circular move is dictated by the amount of centripetal acceleration that the mechanism can accurately sustain. For a typical case of 0.3 g's of acceleration, the robot should be able to negotiate the circle at an average peak speed of 0.345m (13.6 inches) per second, not including the portions required for longitudinal acceleration and deceleration.
   [0010]   In a second case, 1000 moves of 2.54×10^{-4}m (0.01 inches) form a stair case so that there is a 90 degree turn
35   between each segment. If it is assumed that the step size should set the desired accuracy of the system, then the 90 degree changes of direction must be accomplished in a single motor step. Again for 0.3 g's of acceleration, and assuming a typical step size of 7.62×10^{-6}m (0.0003 inches), it is found that the velocity at the corners is maximally 3.35×10^{-3}m (0.132 inches) per second. Even assuming that the robot accelerates and decelerates as fast as possible on the segments between the corners, the maximum velocity on average is 2.67×10^{-2}m (1.05 inches) per second, more than
40   an order of magnitude slower than for the first case.
   [0011]   Clearly, it is necessary to look ahead a considerable distance to determine the nature of the trajectory. If there are sharp turns coming, the velocities should be sufficiently reduced so that acceleration limits are not exceeded. It is also desirable to use the acceleration capabilities of the robot to the greatest extent possible, which means that the robot should preferably always either be accelerating or decelerating.
45   [0012]   Several known types of high performance robotic controllers measure instantaneous position, velocity, and acceleration, and use this information with feed forward corrections based on the coming trajectory to control straight line point-to-point moves. These real time controllers use the fastest available digital signal processor hardware, but are constrained to look, at most, two point-to-point moves ahead. As was indicated in the foregoing example, such a short look-ahead approach can require a significant transit time over a trajectory, as compared to a trajectory that has been
50   globally maximized.

SUMMARY OF THE INVENTION

   [0013]   It is an object of this invention to accurately predict how the kinematics of a robot will dictate the detailed char-
55   acteristics of a time optimized trajectory, and to then use a simple, time based controller to generate the actuator pulse trains required to realize the trajectory.
   [0014]   It is a further object of this invention to provide a method for accomplishing trajectory optimization before the execution of a robot movement.

[0015] Another object of this invention is to provide a method that enables a look-ahead length of thousands of straight line point-to-point moves, using currently available microprocessor technology, to yield a globally optimized trajectory.

[0016] A further object of this invention is to provide a method for pre-processing a desired trajectory to optimize motor drive sequences so as to traverse the trajectory as rapidly as is possible, and without exceeding the mechanical contraints of the underlying robotics system.

[0017] The foregoing and other problems are overcome and the objects of the invention are realized by a method for determining a globally optimal trajectory and velocity for a motor-driven robot. The method utilizes a deflection angle calculation of a maximal velocity for each vertex, a heap sort for globally ensuring that none of the vertices have an excessive velocity, and a vertex adding technique that ensures that the robot is performing straight line moves as rapidly as is possible.

[0018] More particularly, this invention provides a method for `reducing a time required for an object to be driven by a plurality of motors over a predetermined trajectory through an n-dimensional space, wherein n is equal to or greater than two. The trajectory is defined by a plurality of vertices each having coordinates that define a location within the n-dimensional space. The method is characterized by the steps of: (a) providing a set of vertices that define a desired trajectory, each vertex being separated from at least one other vertex by a line segment; (b) for each vertex, determining a maximum velocity that can be sustained while translating the object through a point in the trajectory that corresponds to the vertex; (c) modifying the vertex list, if required, to achieve a velocity profile over the trajectory that causes the object to traverse the trajectory in the shortest period of time, without exceeding the maximum velocity at any vertex; and (d) translating the modified vertex list into motor drive signals for causing the plurality of motors to translate the object in accordance with the modified vertex list.

[0019] In an open loop robotics system, that operates in accordance with the method, the motors are stepper motors that move a step distance of d_j in response to drive signal pulse, where J is the number of degrees of freedom, and where $1 \leq j \leq J$. Translating the object through a point in the trajectory from a vertex i-1 to i to i+1 is shown to result in a deflection of angle θ_i. An aspect of this invention employs a value of $\cos\theta_i$ to determine a delay time between drive signal pulses for a stepper motor associated with an axis having a fastest movement, also referred to as a master axis.

[0020] The step of modifying includes the steps of: (a) determining a velocity profile over a run length corresponding to the desired trajectory, the velocity profile being determined in accordance with a dynamic characteristic of the stepper motor, such as torque; (b) testing an effect of maximum accelerations and maximum decelerations between pairs of vertices to determine if a resulting velocity would exceed the determined maximum velocity at one of the vertices of the pair; and (c) adjusting the velocities of vertices to enable the use of maximum accelerations and decelerations between vertices, and inserting, if required, one or more intermediate vertices between the vertices of the pair.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] The above set forth and other features of the invention are made more apparent in the ensuing Detailed Description of the Invention when read in conjunction with the attached Drawings, wherein:

Fig. 1a is a block diagram of a rapid prototyping system that includes a robot that positions a fabrication tool in accordance with this invention;

Fig. 1b is a block diagram of a further embodiment of the invention;

Fig. 2 graphically depicts a velocity determination technique employed by the invention;

Fig. 3 graphically illustrates an exemplary desired trajectory for a robot, the trajectory including a plurality of vertices;

Fig. 4 is a plot of computed vertex velocities;

Fig. 5 graphically depicts a stepper motor torque curve;

Fig. 6 graphically depicts a resulting velocity profile for the torque curve of Fig. 5;

Fig. 7 plots velocity indices against run length;

Fig. 8 illustrates a revised plot of velocity indices against run length, after applying a step of the flowchart of Fig. 10;

Fig. 9 illustrates a further revised plot of velocity indices against run length, after adding vertices in accordance with

the flowchart of Fig. 10; and

Figs. 10a and 10b depict a flowchart illustrating a presently preferred method of the invention.

5 DETAILED DESCRIPTION OF THE INVENTION

[0022] Figure 1a is a block diagram that illustrates the use of the invention in a rapid prototyping system 10. A robot is generally indicated by the numeral 1. The robot 1 includes a real time controller 12 having outputs 12a-12d coupled to a plurality of stepper motors (SM1-SM4). The stepper motors operate in a conventional fashion so as to provide an 10 incremental rotation (step) of a shaft in response to an input pulse. The stepper motors can also be of the type that provide an incremental linear motion of an actuator element in response to an input pulse. The stepper motors are mechanically coupled to a tool 14 that is used in the rapid prototyping process. By example, the tool 14 is an extrusion nozzle that is coupled to a pump 15 which provides a source of extrudate, such as a polymer. The end result is the controlled deposition of beads 16 of extrudate so as to form a three dimensional model of an object of interest. The illus- 15 trated system is a four degree of motion system wherein the tool 14 is translated within an x-y-z coordinate reference frame by the action of the stepper motors SM1-SM3, plus the extrusion rate of the nozzle controlled by SM4. It should be realized that one or more of the stepper motors can be mechanically coupled to a support structure (not shown) that supports the object during the fabrication thereof. In this type of system the tool 14 may be provided with several degrees of freedom, as may be underlying support structure.

20 [0023] The step size along each axis defines the accuracy to which the position along that axis will be determined (± ½ of the step size). The deflection angle calculation assumes errors of the order of ½ step size are acceptable so that finite deflection angles can be navigated at reasonable velocities. If, instead of a stepper motor, it is desirable to use a synchroneous motor and encoder, the following analysis will also apply as long as 2X the positioning error of the closed loop system is used instead of the step size of the stepper motor.

25 [0024] If a microstepping stepper motor driver is used, then subsequent references to a "step" should be interpreted as "microstep".

[0025] It is pointed out that the robot 1 is an open loop system and does not require the use of positional or other types of feedback to control the positioning of the tool 14.

[0026] It should also be appreciated that the use of an extrusion nozzle for the tool 14 is exemplary only, in that the 30 tool 14 could comprise, by example, a laser or a means for directing a laser beam so as to cure a photopolymer, as is accomplished in stereolithography prototyping systems. Also by example, the tool 14 could be a device, such as a laser or a nozzle for directing a flow of abrasive material, so as to selectively remove material from a block of material.

In general, the teaching of this invention can be applied with advantage in any robotic system where performance is determined by navigating a complex path quickly (at non-uniform velocity). Such applications include flatbed and drum 35 plotters, maze following, glue and caulk applications, film patterning, and solder paste extrusion. The teaching of this invention may also be used, although with less benefit, in applications requiring constant feed rates, such as machining and painting.

[0027] That is, the teaching of the invention is applicable to three dimensional prototyping systems that operate by additive or subtractive methods. Furthermore, the teaching of this invention is applicable to applications other than rapid 40 prototyping systems, particularly applications wherein it is desired to controllably position and/or translate an object.

[0028] In accordance with the teaching of this invention, a processor 18 has an output 18a coupled to an input of the controller 12 and provides a vertex list that is optimized to most efficiently utilize the robot 1. The operation of the processor 18 is described in detail below. The processor 18 has an input 18b for receiving a vertex list from a source, such as an interactive Computer Aided Design (CAD) system. The vertex list that is output by the CAD system may be a 45 specification of at least an outer shell, and possibly also internal support structures, of a three dimensional object to be fabricated by the robot 1. The generation of such vertex lists, based upon an operator's input, is well known in the art, and will not be further described in detail. In general, the vertex list specifies the desired motion of the system by a sequence of straight line segments from one vertex to the next.

[0029] What is now described is a presently preferred method for determining the least time motion of the stepper 50 motor driven robot 1 through a given trajectory.

[0030] The method assumes that the following conditions exist.

1. A set of vertices or points in 2 space, 3 space, or higher dimensionality (additional degrees of freedom) are given that define a desired piecewise linear trajectory for an object: $\bar{V}_i = (x_i, y_i, z_i, p_i, ...) \ 1 \leq i \leq N$.

55

2. A spatial position of an object is manipulated by stepper motors with known dynamic characteristics, such as torque curves and time responses.

4

3. The mechanical characteristics of the robotic manipulator are not time dependent.

[0031] A first goal is to cause the robot 1 to traverse the desired trajectory as quickly as possible. A second goal is to accomplish this process as inexpensively as possible. This second goal suggests an open loop (no feed back) stepper motor system. To maintain the computational requirements to a minimal level, a related goal is to predetermine an optimal control sequence (i.e., which stepper motors to pulse at what times), allowing the inexpensive real-time controller 12 to execute the predetermined sequence.

[0032] A presently preferred embodiment of the method is shown in the flowchart of Figs. 10a and 10b.

[0033] The method operates to reduce a time required for an object, such as an extrusion nozzle, to be driven by a plurality of motors (such as SM1-SM4) over a predetermined trajectory through an n-dimensional space, wherein n is equal to or greater than two. The trajectory is defined by a list of vertices each having coordinates that define a location within the n-dimensional space, each vertex being separated from at least one other vertex by a line segment. A vertex pair is considered to be two vertices that are nearest neighbors along the trajectory and that are separated by a single straight line segment.

[0034] The method executes the following steps:

(A) determining a maximum velocity that can be maintained for each vertex as the object changes direction when being driven from one line segment to the next;

(B) sorting the vertices by ascending maximum velocity;

(C) examining a pair of vertices consisting of the slowest vertex and a subsequent vertex in the trajectory;

(Da) if the determined maximum velocity at the subsequent vertex is found to be exactly achieved by accelerating as fast as possible from the slowest vertex,

(Db) marking the determined maximum velocity at the subsequent vertex as correct;

(Dc) if the determined maximum velocity at the subsequent vertex is found to be greater than the determined maximum velocity,

(Dd) reducing the maximum velocity for the subsequent vertex to the value achieved by accelerating as fast as possible from the slowest vertex; and

(De) marking the reduced maximum velocity at the subsequent vertex as correct.

[0035] If the determined maximum velocity at the subsequent vertex is found to be less than the determined maximum velocity, a next step

(Df) marks the subsequent vertex for a subsequent addition of at least one intermediate vertex.

[0036] The method then examines a pair of vertices consisting of the slowest vertex and a previous vertex in the trajectory.

(Ea) If the determined maximum velocity at the slowest vertex is found to be exactly achieved by decelerating as fast as possible from the previous vertex to the slowest vertex, the method executes the steps of,

(Eb) marking the maximum velocity at the previous vertex as correct;

(Ec) if the determined maximum velocity at the previous vertex is found to be greater than the determined maximum velocity,

(Ed) reducing the maximum velocity for the previous vertex to the value achieved by decelerating as fast as possible from previous vertex to the slowest vertex; and

(Ee) marking the reduced maximum velocity of the previous vertex as correct;
if the determined maximum velocity at the subsequent vertex is found to be less than the determined maximum velocity,

(Ef)   marking the previous vertex for a subsequent addition of at least one intermediate vertex.

(F)   A next step removes the vertex having the lowest maximum velocity from the vertex list; and

(G)   determines if all vertex pairs have been examined.
       If NO at step G, the method repeats the steps (B) through (G); else
       if YES at step G, the method

(H)   adds at least one intermediate vertex between each vertex pair previously marked for vertex addition at a velocity corresponding to maximally accelerating from the previous vertex, and deccelerating towards the subsequent vertex; and

(I)   determines from the resulting vertex list a sequence of motor control signals that will cause the object to be driven over the trajectory so as to achieve the indicated velocities at each vertex of the resulting vertex list.

[0037] That is, the resulting vertex list is downloaded from the processor 18, via cable 18a, to the controller 12, which then determines from the vertex list the stepper motor pulse trains for SM1-SM4 to achieve the optimum motion of the robot 1 and, hence, the fabrication tool 14.

[0038] In that in Fig. 1a the motors are stepper motors, step (H) adds a single vertex if a full acceleration from the first of the pair of vertices, and a full deceleration to the second of the pair of vertices, results in a position, along the line segment connecting the pair of vertices, where the accelerating and decelerating velocities are the same, and which is an integer number of stepper motor steps from both vertices, and where the velocity at the intersection is less than a maximum allowable velocity.

[0039] If a full acceleration, followed by full deceleration, along the line segment between the pair of vertices will cause the velocity to reach a maximum allowable velocity, the step (H) adds a first vertex at a point on the line segment where the velocity first reaches the maximum allowable value, and a second vertex at a point on the line segment whereat a full acceleration will result in the second vertex being reached at the determined maximum velocity.

[0040] If a full acceleration ramp and a full deceleration ramp intersect at a non-integer number of stepper motor steps, the step (H) adds two vertices along the line segment that are separated by one stepper motor step that results in neither an acceleration or a deceleration.

[0041] The real time stepper motor controller 12 must know at what rate and in what direction to pulse each motor SM1-SM4 during the entire trajectory. While not necessary, it will be assumed in what follows that the controller 12 performs a software delay to determine the time between the pulses to the axis requiring the most numbers of steps over a particular segment (referred to herein as the dominant axis for that segment). It is further assumed that pulses to the other axes are produced coincident with the dominant axis pulses, and uniformly distributed along the segment. This is done by interpolation and truncation (see haul ()).

[0042] An important feature in the implementation of the method is the use of a heap sort to find the next lowest velocity vertex, and to re-sort the two nearest neighbor vertices if their velocities are altered. For N vertices, the vertex list is re-sorted maximally N-2 times (the number sorted decreases by one with each sort). As a result, it is desirable that the sorting process be efficient. The heap sort provides a log (N) sort time, and identifies the lowest velocity remaining vertex (the top of the heap). A suitable heap sort algorithm is disclosed by R. Sedgewick, Algorithms, Addison-Wesley, New York, 1988, ISBN 0-201-06673-4, pp. 132-139.

[0043] Another important time-saving feature in the implementation of the method is an efficient and reliable means to determine the maximum velocity associated with a given vertex.

[0044] In this regard, it is assumed that the Pythagorean length of both segments adjoining a vertex has been predetermined. For a rapid prototyping application this is generally true because the length of each segment determines an amount of material to be added or removed. For a rapid prototyping system having an extrusion head, this implies that a required number of extrusion pump steps has been predetermined for each segment connected at a given vertex.

[0045] It is noted that, for 3% rms accuracy,

$$max = maximum\ of\{\ |x|,|y|,|z|\ \}$$

$$\sqrt{x^2 + y^2 + z^2} \approx \frac{4373(max) + 3062(|x| + |y| + |z|)}{2^{13}}$$

[0046] It is also assumed that each axis is independent, and that the time response of each axis is approximately the same. The numbers appearing in the numerator are derived from a numerical fit.

[0047] If axis i moves a distance d_i for each step, and if the average acceleration from a standing start for axis i is a_i, a relationship for the time required to take the first step is given by:

$$\tau_i = \sqrt{\frac{2d_i}{a_i}}.$$

[0048] While the loads and forces on each axis may be different, the step distance d_i can be adjusted for each axis so that the response time τ_i is approximately the same for all axes.

[0049] Using these assumptions, the velocity associated with each vertex can be determined as shown in Fig. 2.

[0050] In moving from vertex i-1 to i to i+1, the trajectory is deflected by θ_i, where

$$\cos\theta_i = \frac{\bar{v}_i \cdot \bar{v}_{i+1}}{|\bar{v}_i| \cdot |\bar{v}_{i+1}|}.$$

[0051] It is assumed that the only acceleration at the vertex is centripetal; i.e. the velocity along the direction of motion is constant

$$|\bar{v}_i| = |\bar{v}_{i+1}| = \frac{d}{\tau},$$

where τ is the minimum time per step for any axis, and d is the step distance of the highest velocity axis. For small deflection' angles, the average centripetal acceleration is

$$2\frac{d}{\tau_i^2}\cos\frac{(\pi - \theta)}{2}$$

solving for the time per step yields:

$$\tau_i = \sqrt{\frac{d}{\sqrt{2}a}}\sqrt[4]{1-\cos\theta}$$

[0052] Since $\sqrt{\frac{d}{a}}$ is the same for all axes, the time delay between pulses is independent of the direction of motion.

[0053] Since the segments are linear, the $\cos(\theta_i)$ can be determined as:

$$\cos(\theta_i) = \frac{x_i \cdot x_{i+1} + y_i \cdot y_{i+1} + z_i \cdot z_{i+1} + p_i \cdot p_{i+1}}{\sqrt{x_i^2 + y_i^2 + z_i^2 + p_i^2}\sqrt{x_{i+1}^2 + y_{i+1}^2 + z_{i+1}^2 + p_{i+1}^2}}.$$

[0054] The determined value of $\cos(\theta_i)$ is preferably used to index into a look-up table to obtain the delay time τ_i between pulses for the fastest moving axis, also referred to as the master axis. It should be noted that the master axis may also be defined to be a virtual axis having an associated movement that is faster than any of the real axes. Clearly, the velocity of vertex i becomes

$$|v_i| = \frac{d_i}{\tau_i}.$$

[0055] It is not necessary for the response times of each axis to be the same, or for the different axes to be independent. The determination of the velocity of a given vertex is similar, but more involved in these cases.

[0056] With respect to the re-setting of vertex velocities, a procedure that guarantees that the robot does not reach any vertex with more velocity than a maximum velocity when going around that vertex is best understood with respect to Fig. 3, which illustrates an exemplary robot trajectory in x-y. For a given segment i, let

$$\text{max}_i = \text{maximum} \{|x_i|, |y_i|, |z_i|, |p_i|\}, \text{ define run } length_i = \sum_{j=1}^{l}$$

5 (similar to the total length traveled). A plot of the computed vertex velocities with respect to run length is shown in Fig. 4.
[0057] A slightly non-standard scale for velocity is used to facilitate later calculations. If the torque curve for a motor is modeled as torque = T_{stall} - K · RPS, as shown in Fig. 5, it is found that the velocity profile is

$$v(t) = v_0 (1-e^{-t/\tau p}),$$

as shown in Fig. 6.
[0058] In the preferred implementation, using a microprocessor for the controller 12, the velocity of an axis is

$$v = d/(n \cdot \tau_d.)$$

where d is the step distance, τ_d is the time for a single software delay loop, and n is the number of delay loops executed.
[0059] A velocity ramp is established as an indexed array of delay loop counts $\{n_i\}$. An acceleration thus involves the use of, by example, n_{j+1} delay loops for a next step. Therefore, only a discrete number of velocities v_j are available:

$$v_j = d/(n_j \tau_d.)$$

25 [0060] The method of denoting velocities by an integer index value and move distances by cumulative master axis steps is useful because an acceleration always appears as a +45° slope line, and similarly a deceleration is a -45° slope line. The object is to provide a velocity index, with respect to run length, of a type shown in Fig. 7, and to therefore connect all vertices with ± 45° sloped lines. In the case shown in Fig. 7, only vertices 9-10 are so connected.
[0061] Fig. 8 shows the result of applying steps C, D and E as described with respect to Figs. 10a and 10b. Vertices
30 that are too fast to be achieved with the nominal acceleration from either nearest neighbor vertices are thus reset to lower velocities.
[0062] Having determined vertices with optimal "correct" velocities, and having reset the velocities of fast vertices, as required, vertex addition is next performed. If a segment is long, compared to the difference between its start vertex velocity and end vertex velocity, time is saved by allowing the robot 1 to first accelerate and then decelerate along the
35 segment. This is equivalent to adding the vertices 11-17, as shown in Fig. 9.
[0063] Because only discrete velocity indices and run lengths are valid, these are cases where the acceleration and deceleration ramps meet at a half-integer.
[0064] In these cases, such as between vertices 5 and 6, two vertices are added (13 and 14) on the nearest integer values.
40 [0065] There are also cases where the acceleration and deceleration ramps would meet at velocities greater than the robot can achieve. This typically will not be the case if the exponential velocity relation (Fig. 6) is used. For this case, two vertices are inserted at the maximum velocity (e.g. vertices 16 and 17).
[0066] The move_compiler subroutine is repeatedly called. Each time the move_compiler subroutine is called a new requested tool 14 position is supplied through global variables (x-requested, y-requested, z-requested). If this is the first
45 call of a sequence, vertices 0-2 are initialized. The requested move is tested for limit violations. If necessary, it is broken up into a series of smaller moves, so that 16-bit integers can be used. The Pythagorean length is computed and material extrusion pump steps are computed, if required. The maximum possible velocity of the previous vertex is then computed. The above procedure is repeated for each call to the move-compiler subroutine until the allocated memory is full, or until the routine indicates the end of the sequence.
50 [0067] Next (after routine dump-it) extra moves to enhance material deposition are added.
[0068] Next, the vertices are sorted by slowest velocity by the heap sort algorithm.
[0069] Next, the nearest neighbor vertices, to the slowest vertex, are tested to see if they are too fast or too slow. If too fast, their velocities are reduced and the heap is re-sorted. The slowest vertex is then removed from the heap and the procedure repeated till the heap is empty.
55 [0070] Next, one or two vertices are added along those segments that can support an acceleration followed by a deceleration.
[0071] Finally, the data is passed to haul (), which generates the stepper motor pulse sequences in real time.
[0072] Fig. 1b illustrates a further embodiment of the invention wherein the operations of the haul() routine are imple-

8

mented in hardware (specifically the block 26).

[0073] In Fig. 1b a robotic system 20 includes a vertex list 22 that is output from a trajectory generator (not shown). An output 22a of the vertex list provides vertices to a move compiler 24 which implements steps of the method as described in detail above. The output 24a of the move compiler 24 is a list comprised of a number of steps for each axis, and also the speed of the master axis. Block 26 implements the functions of the haul() routine and interpolates synchronized pulse trains for each of the axes. The output of the block 26 is provided to a feedback block 28 having an output 28a that is provided to a stepper motor driver block 30. The stepper motor driver block 30 has a plurality (n) outputs for driving stepper motors 1-n.

[0074] The system 20 further include n encoders, one for each stepper motor, having outputs that are provided to an encoder data block 32. The block 32 digitizes the encoder outputs and provides this data as in input 28b to the feedback block 28. Each encoder output is representative of a position of the associated stepper motor.

[0075] In general, and as is indicated by the switch SW1, the system 20 operates in an open loop (OL) configuration when accelerating one or more of the stepper motors, and operates in a closed loop (CL) configuration if it becomes necessary to reposition one or more of the stepper motors. Closed loop operation may be necessary if, for example, a stepper motor were to stall while accelerating or decelerating.

[0076] In operation, the feedback block 28 compares the expected position of each axis with its actual position, as indicated by the associated encoder data. If the two positions are different, and if the axis is not a master axis, and if the step rate of the axis is below a preset threshold, then correction steps are output by the feedback block 28 to restore the errant axis to the desired position.

[0077] The interpolation block 26 may be implemented as an Application Specific Integrated Circuit (ASIC) which performs the functions of the haul() routine in a rapid manner. The blocks 22 and 24 may be contained within an attached processor, such as a workstation, and the link 24a is then a communication link between the workstation and the interpolation block 26 of the actual robot. The feedback block 28 may be implemented with a separate processor that performs the functions of outputting stepper motor drive signals, comparing expected versus actual positions, and generating correcting step sequences when required.

[0078] As was noted previously, the teaching of this invention is not limited to rapid prototyping systems, and is not limited for use only with stepper motors. For Example, the teaching of the invention can be employed with synchronous motors wherein each motor is provided with closed loop control through the use of an encoder or similar feedback sensor. In this case the minimum resolvable distance, 0,0127 mm for example (one half mil), is employed as a "step size", and the command stream output by the move compiler is used an an input to each synchronous motor/encoder combination. The use of the invention enables the closed loop gain to be decreased, thereby improving stability, while providing the benefit of preprocessing the required trajectories to enable rapid and globally optimal positioning.

Claims

1. A method for reducing a time required for an object to be driven by a plurality of motors over a predetermined trajectory through an n-dimensional space, wherein n is equal to or greater than two, the trajectory being defined by a plurality of vertices each having coordinates that define a location within the n-dimensional space, characterized by the steps of:

    providing a set of vertices that define a desired trajectory, each vertex being separated from at least one other vertex by a line segment;

    for each vertex, determining a maximum velocity that can be sustained while translating the object through a point in the trajectory that corresponds to the vertex;

    modifying the vertex list, if required, to achieve a velocity profile over the trajectory that causes the object to traverse the trajectory in the shortest period of time, without exceeding the maximum velocity at any vertex; and

    translating the modified vertex list into motor drive signals for causing the plurality of motors to translate the object in accordance with the modified vertex list.

2. The method as set forth in claim 1, wherein the motors are stepper motors that move a step distance of d in response to drive signal pulse, wherein translating the object through a point in the trajectory from a vertex i-1 to i to i+1 results in a deflection of angle θ_i, and wherein a value of $\cos\theta_i$ is employed to determine a delay time between drive signal pulses for a stepper motor associated with an axis having a fastest movement.

3. The method as set forth in claim 2, wherein the step of modifying includes the steps of:

determining a velocity profile over a run length corresponding to the desired trajectory, the velocity profile being determined in accordance with a dynamic characteristic of the stepper motor;

testing an effect of maximum accelerations and maximum decelerations between pairs of vertices to determine if a resulting velocity would exceed the determined maximum velocity at one of the vertices of the pair; and

adjusting the velocities of vertices to enable the use of maximum accelerations and decelerations between vertices, and inserting, if required, one or more intermediate vertices between the vertices of the pair.

4. The method as set forth in claim 3 wherein the dynamic characteristic is torque.

5. The method of claim 1, comprising the steps of:

(a) determining a maximum velocity that can be maintained for each vertex as the object changes direction when being driven from one line segment to the next;

(b) sorting the vertices by ascending maximum velocity;

(c) examining a pair of vertices consisting of the slowest vertex and a subsequent vertex in the trajectory; if the determined maximum velocity at the subsequent vertex is found to be exactly achieved by accelerating as fast as possible from the slowest vertex,

(d) marking determined maximum velocity at the subsequent vertex as correct; if the determined maximum velocity at the subsequent vertex is found to be greater than the determined maximum velocity,

(e) reducing the maximum velocity for the subsequent vertex to the value achieved by accelerating as fast as possible from the slowest vertex; and

(f) marking the reduced maximum velocity at the subsequent vertex as correct; if the determined maximum velocity at the subsequent vertex is found to be less than the determined maximum velocity,

(g) marking the subsequent vertex for a subsequent addition of at least one intermediate vertex;

(h) examining a pair of vertices consisting of the slowest vertex and a previous vertex in the trajectory; if the determined maximum velocity at the slowest vertex is found to be exactly achieved by decelerating as fast as possible from the previous vertex to the slowest vertex,

(i) marking the maximum velocity at the previous vertex as correct; if the determined maximum velocity at the previous vertex is found to be greater than the determined maximum velocity,

(j) reducing the maximum velocity for the previous vertex to the value achieved by decelerating as fast as possible from previous vertex to the slowest vertex; and

(k) marking the reduced maximum velocity of the previous vertex as correct; if the determined maximum velocity at the subsequent vertex is found to be less than the determined maximum velocity,

(l) marking the previous vertex for a subsequent addition of at least one intermediate vertex;

(m) removing the vertex having the lowest maximum velocity from the vertex list;

(n) determining if all vertex pairs have been examined;

(o) if NO, repeating the steps (b) through (n); else

(p) if YES, adding at least one intermediate vertex between each vertex pair previously marked for vertex addition; and

(q) determining from the resulting vertex list a sequence of motor control signals so as to cause the object to be driven over the trajectory so as to achieve the indicated velocities at each vertex of the resulting vertex list.

6. The method as set forth in claim 5, wherein the motors are stepper motors, and wherein the step (q) adds a single vertex if a full acceleration from the first of the pair of vertices, and a full deceleration to the second of the pair of vertices, results in a position, along the line segment connecting the pair of vertices, where the accelerating and decelerating velocities are the same, and which is an integer number of stepper motor steps from both vertices, and where the velocity at the intersection is less than a maximum allowable velocity.

7. The method as set forth in claim 5, wherein if a full acceleration, followed by full deceleration, along the line segment between the pair of vertices will cause the velocity to reach a maximum allowable velocity, the step (q) adds a first vertex at a point on the line segment where the velocity first reaches the maximum allowable value, and a second vertex at a point on the line segment whereat a full acceleration will result in the second vertex being reached at the determined maximum velocity.

8. The method as set forth in claim 7, wherein the motors are stepper motors, and wherein if a full acceleration ramp and a full deceleration ramp intersect at a non-integer number of stepper motor steps, the step (q) adds two vertices along the line segment that are separated by one stepper motor step that results in neither an acceleration or a deceleration.

9. The method as set forth in claim 5, wherein the step (b) is accomplished with a heap sort.

10. The method as set forth in claim 5, wherein the motors are stepper motors, wherein translating the object through a point in the trajectory from a vertex i-1 to i to i+1 results in a deflection of angle θ_i, and wherein step (a) employs a value of $\cos\theta_i$ to determine the maximum velocity for the vertex i.

Patentansprüche

1. Verfahren zum Reduzieren einer Zeit, die erforderlich ist, um ein Objekt durch mehrere Motoren auf einer vorbestimmten Bahn in einem n-dimensionalen Raum anzutreiben, wobei n gleich oder größer als zwei ist und wobei die Bahn durch eine Vielzahl von Scheitelpunkten definiert ist, die jeweils Koordinaten haben, weiche einen Ort innerhalb des n-dimensionalen Raums definieren, gekennzeichnet durch die Schritte:

Bereitstellen einer Schar von Scheitelpunkten, die eine gewünschte Bahn definieren, wobei jeder Scheitelpunkt von wenigstens einem anderen Scheitelpunkt durch einen Streckenabschnitt getrennt ist;

für jeden Scheitelpunkt, Bestimmen einer maximalen Geschwindigkeit, die ausgeführt werden kann, während sich das Objekt durch einen Punkt auf der Bahn bewegt, welcher dem Scheitelpunkt entspricht;

Modifizieren der Scheitelpunktliste, bei Bedarf, um ein Geschwindigkeitsprofil über der Bahn zu erzielen, welches bewirkt, daß das Objekt die Bahn in der kürzesten Zeitspanne zurücklegt, ohne die maximale Geschwindigkeit in irgendeinem Scheitelpunkt zu überschreiten; und

Umsetzen der modifizierten Scheitelpunktliste in Motoransteuersignale, welche bewirken, daß die Motoren das Objekt gemäß der modifizierten Scheitelpunktliste verschieben.

2. Verfahren nach Anspruch 1, wobei die Motoren Schrittmotoren sind, die sich um eine Schrittdistanz d aufgrund eines Ansteuersignalimpulses bewegen, wobei das Verschieben des Objekts durch einen Punkt auf der Bahn von einem Scheitelpunkt i-1 bis i bis i+1 in einer Ablenkung um einen Winkel θ_i resultiert und wobei ein Wert von $\cos\theta_i$ benutzt wird, um eine Verzögerungszeit zwischen Ansteuersignalimpulsen für einen Schrittmotor zu bestimmen, der einer Achse zugeordnet ist, die eine schnellste Bewegung hat

3. Verfahren nach Anspruch 2, wobei der Schritt des Modifizierens die Schritte beinhaltet:

Bestimmen eines Geschwindigkeitsprofils über einer Lauflänge` die der gewünschten Bahn entspricht wobei das Geschwindigkeitsprofil gemäß einer dynamischen Eigenschaft des Schrittmotors bestimmt wird;

Testen einer Auswirkung von maximalen Beschleunigungen und maximalen Verlangsamungen zwischen Paaren von Scheitelpunkten, um festzustellen ob eine resultierende Geschwindigkeit die vorbestimmte Maximalgeschwindigkeit in einem der Scheitelpunkte des Paares übersteigen würde; und

Einstellen der Geschwindigkeiten von Scheitelpunkten, um die Benutzung von maximalen Beschleunigungen und Verlangsamungen zwischen Scheitelpunkten zu ermöglichen, und Einfügen, bei Bedarf, von einem oder mehreren Zwischenscheitelpunkten zwischen den Scheitelpunkten des Paares.

4. Verfahren nach Anspruch 3, wobei die dynamische Eigenschaft das Drehmoment ist.

5. Verfahren nach Anspruch 1, beinhaltend die Schritte:

(a) Ermitteln einer maximalen Geschwindigkeit, die für jeden Scheitelpunkt aufrechterhalten werden kann, wenn das Objekt die Richtung ändert, wenn es von einem Streckenabschnitt zum nächsten angetrieben wird;

(b) Sortieren der Scheitelpunkte nach aufsteigender maximaler Geschwindigkeit;

(c) Prüfen von einem Paar Scheitelpunkten, das aus dem langsamsten Scheitelpunkt und einem anschließenden Scheitelpunkt auf der Bahn besteht;
wenn es sich zeigt, daß die ermittelte maximale Geschwindigkeit in dem anschließenden Scheitelpunkt exakt erreicht wird, indem so schnell wie möglich von dem langsamsten Scheitelpunkt aus beschleunigt wird,

(d) Markieren der ermittelten maximalen Geschwindigkeit in dem anschließenden Scheitelpunkt als korrekt;
wenn es sich zeigt, daß die ermittelte maximale Geschwindigkeit in dem anschließenden Scheitelpunkt größer als die ermittelte maximale Geschwindigkeit ist,

(e) Reduzieren der maximalen Geschwindigkeit für den anschließenden Scheitelpunkt auf den Wert, der erzielt wird, indem von dem langsamsten Scheitelpunkt aus so schnell wie möglich beschleunigt wird;

(f) Markieren der reduzierten maximalen Geschwindigkeit in dem anschließenden Scheitelpunkt als korrekt;
wenn es sich zeigt, daß die ermittelte maximale Geschwindigkeit in dem anschließenden Scheitelpunkt niedriger als die ermittelte maximale Geschwindigkeit ist,

(g) Markieren des anschließenden Scheitelpunktes für eine anschließende Addition von wenigstens einem Zwischenscheitelpunkt;

(h) Prüfen von einem Paar Scheitelpunkten` das aus dem langsamsten Scheitelpunkt und einem vorhergehenden Scheitelpunkt auf der Bahn besteht;
wenn es sich zeigt, daß die ermittelte maximale Geschwindigkeit in dem langsamsten Scheitelpunkt exakt erreicht wird, indem so schnell wie möglich ab dem vorhergehenden Scheitelpunkt bis zu dem langsamsten Scheitelpunkt verlangsamt wird,

(i) Markieren der maximalen Geschwindigkeit in dem vorhergehenden Scheitelpunkt als korrekt;
wenn es sich zeigt, daß die ermittelte maximale Geschwindigkeit in dem vorhergehenden Scheitelpunkt größer als die ermittelte maximale Geschwindigkeit ist,

(j) Reduzieren der maximalen Geschwindigkeit für den vorhergehenden Scheitelpunkt auf den Wert, der erreicht wird, indem so schnell wie möglich ab dem vorhergehenden Scheitelpunkt bis zu dem langsamsten Scheitelpunkt verlangsamt wird; und

(k) Markieren der reduzierten maximalen Geschwindigkeit des vorhergehenden Scheitelpunktes als korrekt;
wenn es sich zeigt, daß die ermittelte maximale Geschwindigkeit in dem anschließenden Scheitelpunkt kleiner als die ermittelte maximale Geschwindigkeit ist,

(l) Markieren des vorhergehenden Scheitelpunktes für eine anschließende Addition von wenigstens einem

Zwischenscheitelpunkt;

(m) Entfernen des Scheitelpunktes' der die niedrigste maximale Geschwindigkeit hat, aus der Scheitelpunktliste;

(n) Feststellen, ob alle Scheitelpunktpaare geprüft worden sind;

(o) wenn NEIN, Wiederholen der Schritte (b) bis (n); sonst

(p) wenn JA, Addieren von wenigstens einem Zwischenscheitelpunkt zwischen jedem Scheitelpunktpaar, das zuvor für die Scheitelpunktaddition markiert worden ist; und

(q) Ermitteln einer Sequenz von Motorsteuersignalen aus der resultierenden Scheitelpunktliste, um so zu bewirken, daß das Objekt so auf der Bahn angetrieben wird, daß die angezeigten Geschwindigkeiten in jedem Scheitelpunkt der resultierenden Scheitelpunktliste erzielt werden.

6. Verfahren nach Anspruch 5, wobei die Motoren Schrittmotoren sind und wobei der Schritt (q) einen einzelnen Scheitelpunkt addiert wenn eine volle Beschleunigung ab dem ersten Scheitelpunkt des Paares von Scheitelpunkten und eine volle Verlangsamung bis zu dem zweiten Scheitelpunkt des Paares von Scheitelpunkten in einer Position längs des Streckenabschnitts resultiert, welcher das Paar von Scheitelpunkten verbindet, wo die Beschleunigungs- und Verlangsamungsgeschwindigkeiten dieselben sind, und welches eine ganze Zahl von Schrittmotorschritten von beiden Scheitelpunkten aus ist, und wo die Geschwindigkeit in dem Schnittpunkt kleiner ist als eine maximal zulässige Geschwindigkeit.

7. Verfahren nach Anspruch 5, wobei, wenn eine volle Beschleunigung, gefolgt von einer vollen Verlangsamung, längs des Streckenabschnitts zwischen dem Paar Scheitelpunkten bewirken wird, daß die Geschwindigkeit eine maximal zulässige Geschwindigkeit erreicht, der Schritt (q) einen ersten Scheitelpunkt in einem Punkt auf dem Streckenabschnitt addiert, wo die Geschwindigkeit zum ersten Mal den maximal zulässigen Wert erreicht, und einen zweiten Scheitelpunkt in einem Punkt auf dem Streckenabschnitt, ab welchem eine volle Beschleunigung dazu führen wird, daß der zweite Scheitelpunkt mit der ermittelten maximalen Geschwindigkeit erreicht wird.

8. Verfahren nach Anspruch 7, wobei die Motoren Schrittmotoren sind, und wobei, wenn eine volle Beschleunigungs-rampe und eine volle Verlangsamungsrampe sich in einer unganzzahligen Zahl von Schrittmotorschritten schneiden, der Schritt (q) zwei Scheitelpunkte längs des Streckenabschnitts addiert, die durch einen Schrittmotorschritt getrennt sind, der weder zu einer Beschleunigung noch zu einer Verlangsamung führt.

9. Verfahren nach Anspruch 5, wobei der Schritt (b) mit einer Gruppensortierung ausgeführt wird.

10. Verfahren nach Anspruch 5, wobei die Motoren Schrittmotoren sind, wobei das Bewegen des Objekts durch einen Punkt auf der Bahn von einem Scheitelpunkt i-1 bis i bis i+1 zu einer Ablenkung um einen Winkel θ_i führt und wobei der Schritt (a) einen Wert von $\cos \theta_i$ verwendet, um die maximale Geschwindigkeit für den Scheitelpunkt i zu ermitteln.

Revendications

1. Procédé pour réduire le temps exigé pour qu'un objet soit entraîné par une pluralité de moteurs sur une trajectoire prédéterminée à travers un espace à n dimensions, n étant égal ou supérieur à deux, la trajectoire étant définie par une pluralité de sommets ayant chacun des coordonnées qui définissent un emplacement dans l'espace à n dimensions, caractérisé en ce qu'il comprend les étapes consistant à fournir un ensemble de sommets qui définissent une trajectoire désirée, chaque sommet étant séparé d'au moins un autre sommet par un segment de ligne, à déterminer, pour chaque sommet, une vitesse maximale qui peut être maintenue lors du passage de l'objet par un point sur la trajectoire qui correspond au sommet, à modifier, si nécessaire, la liste des sommets pour obtenir un profil de vitesse, sur la trajectoire, qui amène l'objet à parcourir la trajectoire dans la plus courte période de temps, sans dépasser la vitesse maximale à un sommet quelconque, et à convertir la liste des sommets modifiée en signaux d'attaque des moteurs pour amener la pluralité de moteurs à déplacer l'objet en accord avec la liste des sommets modifiée.

2. Procédé suivant la revendication 1 caractérisé en ce que les moteurs sont des moteurs pas à pas qui se déplacent

d'une distance de pas d en réponse à une impulsion de signal d'attaque, le passage de l'objet par un point sur la trajectoire, d'un sommet il à un sommet i puis à un sommet i+1, se traduit par une déviation d'un angle θ_i, et une valeur de cos θ_i est employée pour déterminer un temps de retard entre des impulsions de signal d'attaque pour un moteur pas à pas associé à un axe ayant le mouvement le plus rapide.

3. Procédé suivant la revendication 2 caractérisé en ce que l'étape de modification comporte les étapes consistant à déterminer un profil de vitesse sur une longueur de course correspondant à la trajectoire désirée, le profil de vitesse étant déterminé en accord avec une caractéristique dynamique du moteur pas à pas, à tester un effet d'accélérations maximales et de décélérations maximales entre des paires de sommets afin de déterminer si une vitesse résultante dépasserait la vitesse maximale déterminée à l'endroit de l'un des sommets de la paire, et à ajuster les vitesses des sommets de manière à permettre l'utilisation d'accélérations et décélérations maximales entre les sommets et a insérer, si nécessaire, un ou plusieurs sommets intermédiaires entre les sommets de la paire.

4. Procédé suivant la revendication 3 caractérisé en ce que la caractéristique dynamique est le couple.

5. Procédé suivant la revendication 1 caractérisé en ce qu'il comprend les étapes consistant :

    (a) à déterminer une vitesse maximale qui peut être maintenue pour chaque sommet tandis que l'objet change de direction lorsqu'il est entraîné d'un segment de ligne au segment suivant;

    (b) à trier les sommets dans l'ordre de leur vitesse maximale croissante;

    (c) à examiner une paire de sommets constituée par le sommet le plus lent et un sommet subséquent dans la trajectoire;

    si la vitesse maximale déterminée à l'endroit du sommet subséquent est trouvée être atteinte exactement en accélérant aussi rapidement que possible à partir du sommet le plus lent,

    (d) à marquer la vitesse maximale déterminée à l'endroit du sommet subséquent comme étant correcte;

    si la vitesse maximale déterminée à l'endroit du sommet subséquent est trouvée être supérieure à la vitesse maximale déterminée,

    (e) à réduire la vitesse maximale pour le sommet subséquent à la valeur atteinte par une accélération aussi rapide que possible à partir du sommet le plus lent ; et

    (f) à marquer la vitesse maximale réduite à l'endroit du sommet subséquent comme étant correcte;

    si la vitesse maximale déterminée à l'endroit du sommet subséquent est trouvée être inférieure à la vitesse maximale déterminée,

    (g) à marquer le sommet subséquent en vue d'une addition subséquente d'au moins un sommet intermédiaire;

    (h) à examiner une paire de sommets constituée par le sommet le plus lent et un sommet précédent dans la trajectoire;

    si la vitesse maximale déterminée à l'endroit du sommet le plus lent est trouvée être atteinte exactement par une décélération aussi rapide que possible à partir du sommet précédent jusqu'au sommet le plus lent,

    (i) à marquer la vitesse maximale à l'endroit du sommet précédent comme étant correcte;

    si la vitesse maximale déterminée à l'endroit du sommet précédent est trouvée être supérieure à la vitesse maximale déterminée,

    (j) à réduire la vitesse maximale pour le sommet précédent à la valeur atteinte par une décélération aussi rapide que possible à partir du sommet précédent jusqu'au sommet le plus lent; et

    (k) à marquer la vitesse maximale réduite du sommet précédent comme étant correcte;

    si la vitesse maximale déterminée à l'endroit du sommet subséquent est trouvée être inférieure à la vitesse maximale déterminée,

    (l) à marquer le sommet précédent en vue d'une addition subséquente d'au moins un sommet intermédiaire;

    (m) à supprimer le sommet ayant la vitesse maximale la plus basse de la liste des sommets;

    (n) à déterminer si toutes les paires de sommets ont été examinées;

    (o) si la réponse est NON, à répéter les étapes (b) à (n);

    (p) si la réponse est OUI, à ajouter au moins un sommet intermédiaire entre chaque paire de sommets précédemment marqués en vue d'une addition d'un sommet; et

    (q) à déterminer, à partir de la liste des sommets résultante, une séquence de signaux de commande des moteurs de manière à amener l'objet à être entraîné suivant la trajectoire afin d'atteindre les vitesses indiquées à l'endroit de chaque sommet de la liste des sommets résultante.

6. Procédé suivant la revendication S caractérisé en ce que les moteurs sont des moteurs pas à pas et l'étape (g) ajoute un seul sommet si une accélération totale à partir du premier sommet de la paire de sommets et une décé-

lération totale vers le second sommet de la paire de sommets se traduit par une position, le long du segment de ligne connectant la paire de sommets, où les vitesses d'accélération et décélération sont les mêmes, et qui est un nombre entier de pas des moteurs pas à pas à partir des deux sommets, et où la vitesse à l'intersection est inférieure à une vitesse maximale admissible.

7. Procédé suivant la revendication 5 caractérisé en ce que si une accélération totale, suivie par une décélération totale, le long du segment de ligne entre la paire de sommets amène la vitesse à atteindre une vitesse maximale admissible, l'étape (q) ajoute un premier sommet à l'endroit d'un point, sur le segment de ligne, où la vitesse atteint en premier la vitesse maximale admissible, et un second sommet en un point, sur le segment de ligne, où une accélération totale se traduit par le fait que le second sommet est atteint à la vitesse maximale déterminée.

8. Procédé suivant la revendication 7 caractérisé en ce que les moteurs sont des moteurs pas à pas et si une rampe d'accélération totale et une rampe de décélération totale se recoupent à l'endroit d'un nombre non entier de pas des moteurs pas à pas, l'étape (q) ajoute, le long du segment de ligne, deux sommets qui sont séparés par un pas des moteurs pas à pas, ce qui se traduit par ni une accélération ni une décélération.

9. Procédé suivant la revendication 5 caractérisé en ce que l'étape (b) est accomplie avec un tri en tas.

10. Procédé suivant la revendication 5 caractérisé en ce que les moteurs sont des moteurs pas à pas, le passage de l'objet par un point sur la trajectoire, d'un sommet i-1 à un sommet i puis à un sommet i+1 se traduit par une déviation d'un angle θ_i, et une valeur de $\cos \theta_i$ est employée dans l'étape (a) pour déterminer la vitesse maximale pour le sommet i.
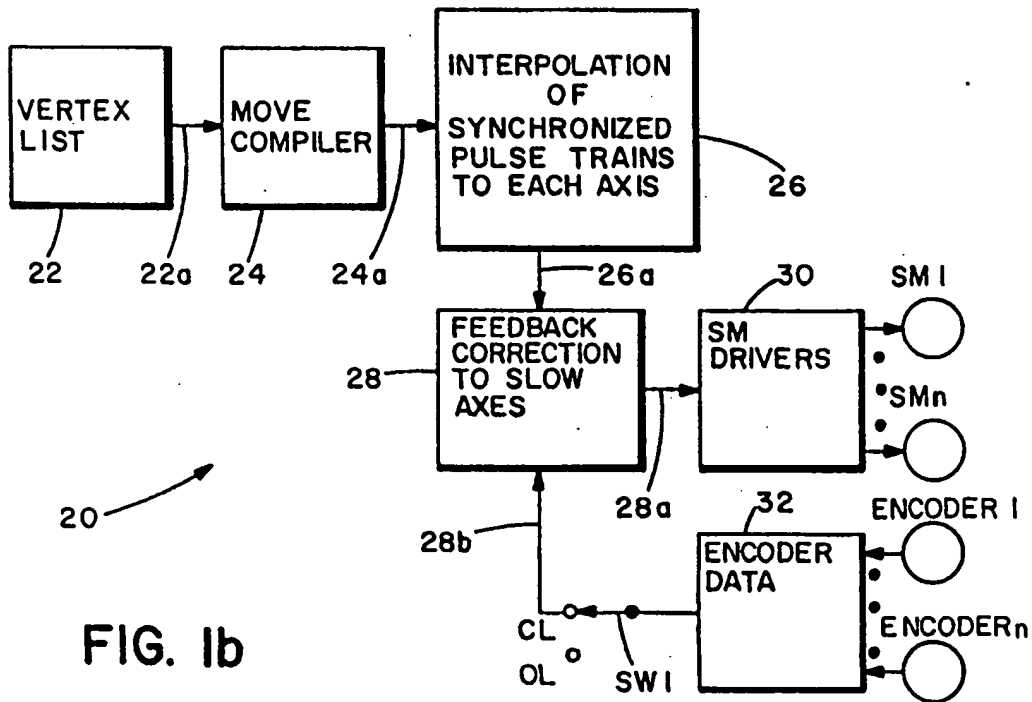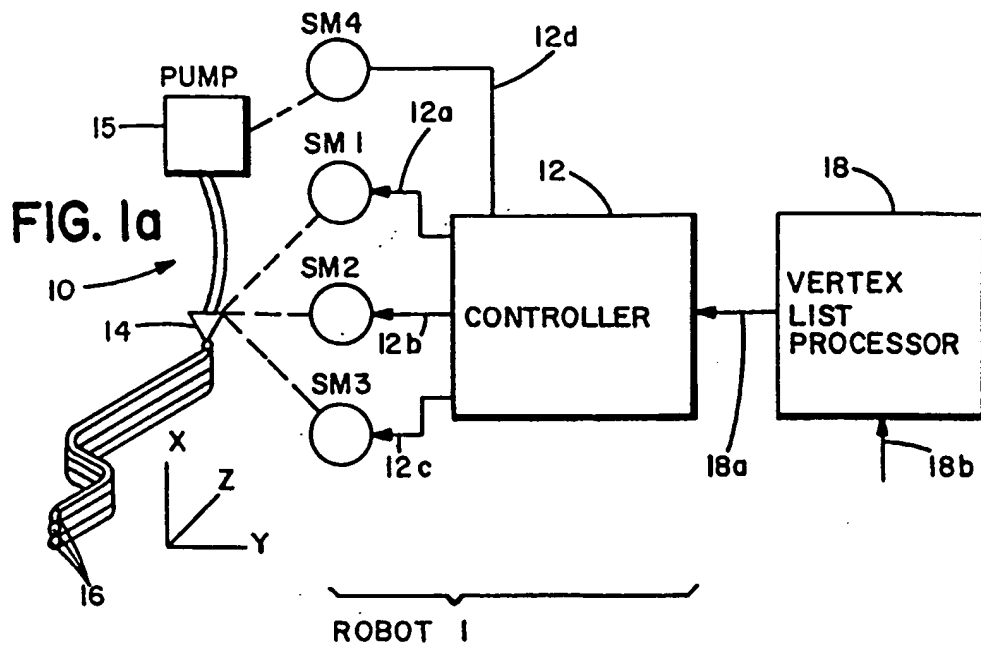
FIG. 1a

PUMP

SM4   12d

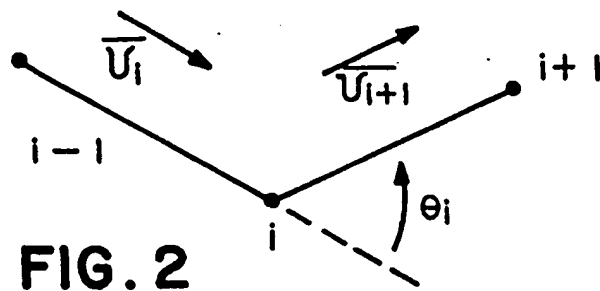SM1   12a

SM2

14   12b

SM3   12c

X

Z

Y

16

10

15

CONTROLLER   12

VERTEX LIST PROCESSOR   18

18a

18b

ROBOT 1

FIG. 1b

VERTEX LIST   22

MOVE COMPILER   24

INTERPOLATION OF SYNCHRONIZED PULSE TRAINS TO EACH AXIS   26

22a   24a   26a

FEEDBACK CORRECTION TO SLOW AXES   28

SM DRIVERS   30   SM1

28a   SMn

32   ENCODER 1

28b   ENCODER DATA

CL

OL   SW1   ENCODERn

20

FIG. 2

FIG. 3

FIG. 4

VERTEX
VELOCITY

RUN LENGTH

TORQUE

$$TORQUE = T_{STALL} - K \cdot RPS$$

RPS

FIG. 5

VELOCITY

$$\upsilon(t) = \upsilon_0(1 - e^{t/\tau_p})$$

TIME

FIG. 6

VELOCITY
INDEX

FIG. 7

1   2   3   4   5   6   7   8   9   10
RUN LENGTH

FIG. 8

FIG. 9

DETERMINE THE VELOCITIES THAT CAN BE MAINTAINED
FOR EACH VERTEX AS THE ROBOT CHANGES
DIRECTION GOING FROM ONE SEGMENT TO THE NEXT — A

B

SORT THE VERTICES BY
ASCENDING VELOCITY — B

CHECK THE SLOWEST VERTEX WITH RESPECT TO ITS
PREVIOUS AND SUBSEQUENT VERTEX IN THE
TRAJECTORY — C

IS
CALCULATED — Da
VELOCITY OF
SUBSEQUENT VERTEX
ACHIEVED BY ACCELERATING   YES
AS FAST AS POSSIBLE
FROM THE SLOWEST
VERTEX ?

NO

IS
CALCULATED — Dc
VELOCITY OF
SUBSEQUENT VERTEX
GREATER THAN THAT   YES   Dd
ACHIEVED BY ACCELERATING   REDUCE SUBSEQUENT
AS FAST AS POSSIBLE   VERTEX TO VALUE
FROM SLOWEST   ACHIEVED BY FULL
VERTEX ?   ACCELERATION

NO

De   MARK AS — Db
                                         CORRECT

MARK VERTEX TO ADD
INTERMEDIATE VERTEX   A

FIG. 10a

FIG. 10b

A

Ea

IS CALCULATED VELOCITY OF PREVIOUS VERTEX ACHIEVED BY DECELERATING AS FAST AS POSSIBLE FROM THE PREVIOUS VERTEX TO THE CURRENT VERTEX ?

YES

NO

Ec

IS CALCULATED VELOCITY OF PREVIOUS VERTEX GREATER THAN THAT ACHIEVED BY DECELERATING AS FAST AS POSSIBLE FROM THE PREVIOUS VERTEX TO THE CURRENT VERTEX ?

YES

Ed

REDUCE PREVIOUS VERTEX TO VALUE ACHIEVED BY FULL DECELERATION

MARK AS CORRECT

Eb

Ee

MARK VERTEX TO ADD INTERMEDIATE VERTEX

F

REMOVE THE CURRENT SLOWEST VERTEX

G

ALL VERTEX PAIRS TESTED?

NO

B

YES

H

ADD 1 OR 2 VERTICES BETWEEN EACH VERTEX PAIR PREVIOUSLY MARKED FOR ADDITION

I

EXECUTE RESULTING VERTEX LIST
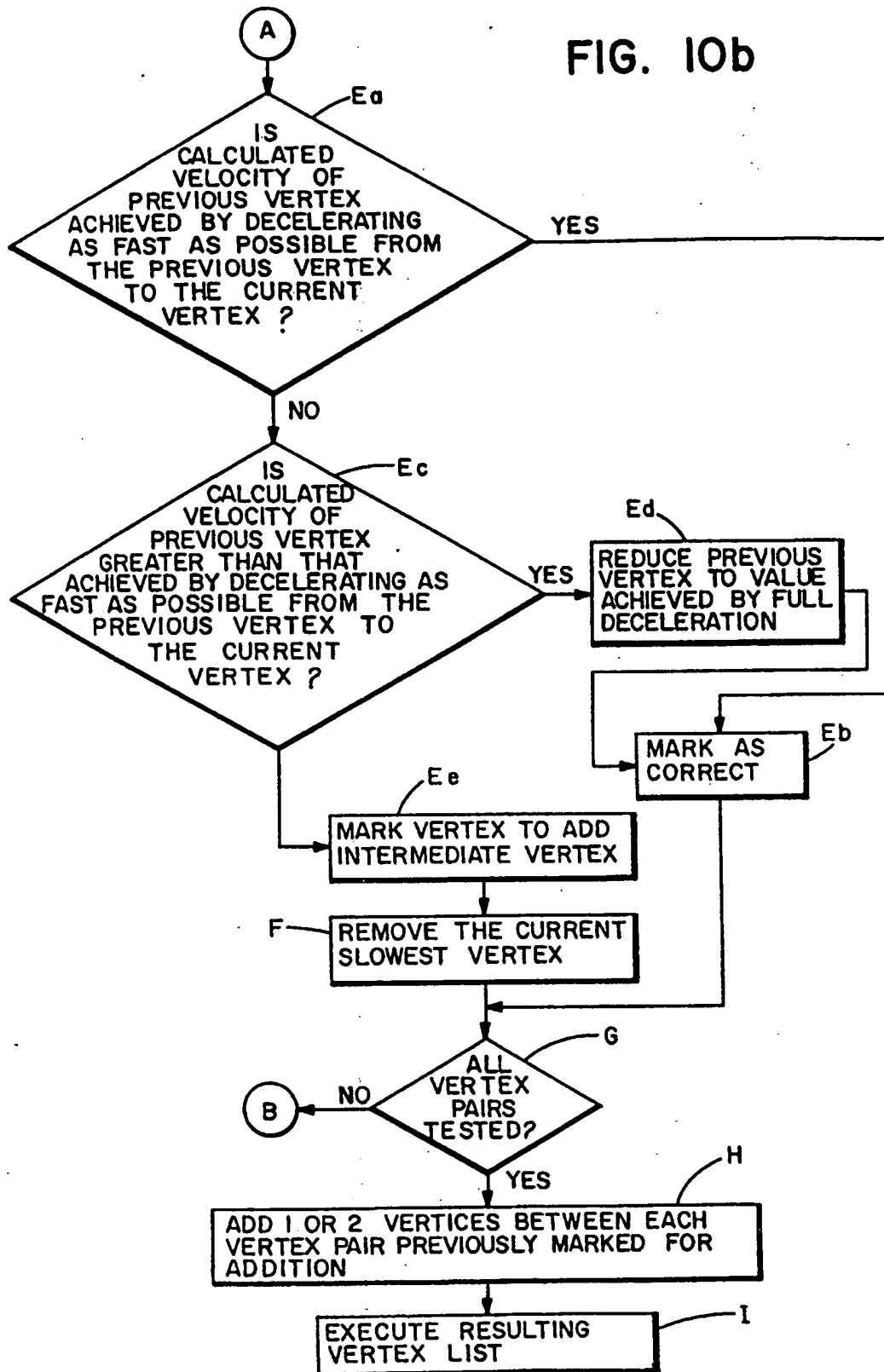
European Patent
Office

**SUPPLEMENTARY
EUROPEAN SEARCH REPORT**

Application Number

EP 89 90 7822

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
X	US-A-4517652 (BENNETT R.L. ET AL.) * column 3, lines 23 - 46 * * column 7, line 53 - column 8, line 14 * * column 13, line 34 - column 15, line 8; figures 1, 2, 11, 28-30 *	1-3	G05B19/42 B25J9/02 B23K9/095
Y		4	
Y	Proceedings of the 1987 International Conference on Robotics & Automation, March 31-April 3, 1987 Raleigh, North Carolina, US vol. 2, 1987, New York, US pages 625 - 630; DeCurtins J. & Kremers J.: "SKETCH : A Simple-to-use Programming System for Visually Guided Robotic Arc Welding" * pages 625 - 630 *	4	

TECHNICAL FIELDS
SEARCHED (Int. Cl.5)

G05B

The supplementary search report has been drawn
up for the claims attached hereto.

Place of search	Date of completion of the search	Examiner
THE HAGUE	23 NOVEMBER 1990	HAUSER L.E.R.

C L A I M S

1. A visually controlled robot system comprising:

a robot control device for controlling an operation of a robot body;

5 a vision sensor system for detecting visual information; and

a common operator panel having means for selecting one of a first mode for carrying out an operation related to said robot control device and a second mode 10 for carrying out an operation related to said vision sensor system, to carry out said operations.

2. A visually controlled robot system according to claim 1, wherein said vision sensor system includes a vision sensor for imaging at least a subject of work, 15 and a vision sensor controller for analyzing a picture signal from said vision sensor, said vision sensor controller being connected to said operator panel.

3. A visually controlled robot system according to claim 1, wherein an operation common to said first 20 and second modes is executed through an identical operation of the operator panel by an operator.

4. A visually controlled robot system according to any one of claims 1 to 4, wherein a welding torch is mounted to said robot body to carry out arc welding.